UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/561,135	04/25/2006	Martin Vorbach	2885/98	9404

26646 7590 07/15/2010

KENYON & KENYON LLP
ONE BROADWAY
NEW YORK, NY 10004

EXAMINER	
ALROBAYE, IDRISS N	
ART UNIT	PAPER NUMBER
2183	
MAIL DATE	DELIVERY MODE
07/15/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/561,135	VORBACH ET AL.	
	Examiner	Art Unit	
	IDRISS N. ALROBAYE	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 May 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-7 and 12-72* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-7 and 12-72* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *16 May 2008* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *11/18/2009; /22/2010; 06/03/2010; 06/29/2010*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. This action is responsive to amendments received on 5/03/2010.

2. Claims 1-7 and 12-72 presented for examination. Claims 8-11 cancelled.

3. Applicant's amendments to the claims to overcome 35 USC 112 rejections have been considered and the rejections have been withdrawn.

4. Note there were two exact copies of the IDS filed on 11/18/2009, thus one of them was not deleted (crossed out).

Claim Objections

5. Claims 2-4, 12-20, 29-56 and 58-72 recite "A method according to claim...", it is suggested to change it to --The method according to claim…-- such that it is clear that the dependent claims are further limiting the independent claims upon which they depend, as oppose to a new method.

6. Claims 21-28 should also be changed from "A device according to claim..." to -- The device according to claim…-- so that is clear that the dependent claims are further limiting the independent claims.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 1-4, 7, 12-20 and 29-72 are rejected under 35 U.S.C. 103(a) as being unpatentable over John Reid Hauser "Augmenting a Microprocessor with reconfigurable hardware" (hereinafter Hauser" in view of Rosenbluth et al. U.S. Patent No. 6,868,476 (hereinafter Rosenbluth).

9. As per claim 1, Hauser teaches a method of processing data comprising the step of:

coupling:

(a) at least one unit adapted for processing data in a sequential manner and comprising an instruction pipeline (see page 11, 'master processor'; see also page 25, Fig. 3.1(b), wherein the pipeline of the processor is sequential); and

(b) an array adapted for processing data (page 25, Fig. 3.1(b) 'reconfigurable array'; see also page 11 'slave reconfigurable device'), the array comprising a plurality of data processing cells (see page 12, Fig. 2.6) having programmable coarse grained arithmetic logic units (ALUs) (see figures in page 62; see figure shown in page 64; see also page 52, section 3.2.3 for more details)

wherein:

the at least one unit is operable independently of the array (see page 11, 'master processor'; see also page 25, Fig. 3.1(b)); and

the array:

processes at least one program that is compiled from a high-level language (the instructions for processing are compiled from a high-level language such as C; see page 20, paragraph beginning with 'Probably the earliest…'; see also figure in page 108 and the 'software tools'); and

is:

at least one of coarse grained (see page 52, section 3.2.3 'array granularity' and figure shown in page 62) and runtime reconfigurable (see page 25, Fig. 3.1(b) and code in page 111, wherein the array is dynamically configurable); and

controlled by instructions issued from the instruction pipeline of the at least one unit to the array (see page 11, last paragraph; see also page 25, Fig. 3.1 (a), wherein the processor controls the array; see also description of the figure in page 37);

Hauser shows network communication between the main processor and the array but did not explicitly show that the array comprises a packet-oriented network communicating values in packets. However, the secondary reference (Rosenbluth) teaches an array (Rosenbluth Fig. 1, element 20's 'MEs') comprises a packet-oriented network communicating values in packets (see Rosenbluth, Fig. 3, packet processing; see also col. 2, lines 36-54), for the purpose of processing network packets between processors or a network of functional units with a maximum efficiency;

Accordingly, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teachings of Rosenbluth in the invention of Hauser, for the purpose of efficiently transmitting packets between functional units or

processors within a network and significantly improving performances when transmitting in packets rather than in single instructions.

10. As per claim 2, Hauser further teaches the method according to claim 1, further comprising the step of:

transferring via at least one data path at least one of an input data and an output data from the at least one unit to the array and from the array to the at least one unit (see Fig. 3.1(b) in page 25, wherein data is transferred between the processor and the array), the at least one data path being provided therebetween and comprising at least one FIFO (see Fig. 3.1(b) wherein a register or a FIFO is between the array and the processor, see also interface in page 37) so as to allow for at least one of (i) a coupling between the at least one unit and the array that is not synchronous and (ii) a data processing within the at least one unit and the array that is not synchronous (see description of the figures in page 11 and page 27).

11. As per claim 3, Hauser further teaches a method according to claim 2, wherein the transferring is performed by at least one of inserting data directly into and extracting data directly from a data path of at least one of the at least one unit and the array (see page 4, paragraph beginning with "if a page miss occurs...", see also description in page 25).

12. As per claim 4, Hauser further teaches a method according to claim 3, further comprising the step of:

providing between the at least one unit and the array a path adapted for transfer of at least one of status information and event information (see page 25, first paragraph and page 82, first paragraph; see also the state of the array in page 218).

13. As per claim 7, Hauser teaches a method of processing data comprising the steps of:

coupling:

(a) at least one unit adapted for processing data in a sequential manner and comprising an instruction pipeline (see page 11, 'master processor'; see also page 25, Fig. 3.1(b), wherein the pipeline of the processor is sequential); and

(b) an array (page 25, Fig. 3.1(b) 'reconfigurable array') that:

is adapted for processing data (page 25, Fig. 3.1(b) 'reconfigurable array'; see also page 11 'slave reconfigurable device'),

comprises a plurality of data processing cells (see page 12, Fig. 2.6) having programmable coarse grained arithmetic logic units (ALUs) (see figures in page 62; see figure shown in page 64; see also page 52, section 3.2.3 for more details)

processes at least one program that is compiled from a high-level language (the instructions for processing are compiled from a high-level language such as C; see page 20, paragraph beginning with 'Probably the earliest…'; see also figure in page 108 and the 'software tools'); and

providing a path allowing for block data transfer between the array and the at least one unit through a data cache (see page 30, paragraph beginning with 'With configuration sizes…" and last paragraph in page 37; see also figure in page 61, wherein both the array and the processor retrieves and write data to the cache);

Hauser shows network communication between the main processor and the array but did not explicitly show that the array comprises a packet-oriented network communicating values in packets. However, the secondary reference (Rosenbluth) teaches an array (Rosenbluth Fig. 1, element 20's 'MEs') comprises a packet-oriented network communicating values in packets (see Rosenbluth, Fig. 3, packet processing; see also col. 2, lines 36-54), for the purpose of processing network packets between processors or a network of functional units with a maximum efficiency;

Accordingly, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teachings of Rosenbluth in the invention of Hauser, for the purpose of efficiently transmitting packets between functional units or processors within a network and significantly improving performances when transmitting in packets rather than in single instructions.

14. As per claim 12, Hauser further teaches a method according to claim 1, wherein the at least one unit includes at least one of a CPU, a von-Neumann-Processor, and a microcontroller (see figure in page 11 and 25).

15. As per claim 13, Hauser further teaches a method according to claim 1, wherein the array includes at least one of a data processor, a Field Programmable Gate-Array (FPGA), a Data Flow Processor (DFP), a Digital Signal Processor (DSP), an eXtreme Processing Platform (XPP), and a chaemeleon-technology data processing fabric (see last two lines in page 41 and section 4.1.1, page 61).

16. As per claim 14, Hauser further teaches a method according to claim 2, wherein the at least one data path between the at least one unit and the array includes at least one local memory connected to the at least one unit as a cache and connected to the array (see figure in page 61 and its description).

17. As per claim 15, Hauser further teaches further teaches a method according to claim 14, wherein the at least one local memory includes an internal RAM (IRAM) (description in page 25 and figure in page 61).

18. As per claim 16, Hauser further teaches further teach a method according to claim 1, wherein configuration information for the array is issued by the instruction pipeline of the at least one unit (see page 11, last paragraph; see also page 25, Fig. 3.1 (a), wherein the processor controls the array; see also description of the figure in page 37).

19. As per claim 17, Hauser further teaches a method according to claim 16, further comprising:

buffering the configuration information in at least one FIFO so as to allow for at least one of a coupling between and a data processing within the at least one unit and the array that is not strictly synchronous (see Fig. 3.1(b) wherein a register or a FIFO is between the array and the processor, see also interface in page 37).

20. As per claim 18, Hauser further teaches a method according to claim 1, wherein the at least one unit supports multi-threading, and the array is connected as a thread unit (see page 18, first bullet).

21. As per claim 19, Hauser further teaches a method according to claim 18, wherein the array is operable synchronously to the unit (see Fig. 3.1(a) in page 25 and its description).

22. As per claim 20, Hauser further teaches a method according to claim 4, wherein the at least one of the status information and the event information includes at least one of flags, an overflow, and a carry (see page 25, first paragraph and page 82, first paragraph; see also the state of the array in page 218).

23. As per claim 29, Hauser further teaches a method according to claim 7, wherein the at least one unit includes at least one of a CPU, a von-Neumann-processor, and a microcontroller (see figure in page 11 and 25).

24. As per claim 30, Hauser further teaches a method according to claim 7, wherein the array includes at least one of a runtime and reconfigurable data processor, a Data Flow Processor (DFP), a Digital Signal Processor (DSP), an eXtreme Processing Platform (XPP), and a chaemeleon-technology data processing fabric (see last two lines in page 41 and section 4.1.1, page 610).

25. As per claim 31, Hauser further teaches a method according to claim 7, wherein the array is controlled by instructions issued from the instruction pipeline of the at least one unit to the array (see page 11, last paragraph; see also page 25, Fig. 3.1 (b) or (a), wherein the processor controls the array; see also description of the figure in page 37).

26. As per claim 32, Hauser further teaches a method according to claim 7, wherein the array is controlled by instructions issued from the instruction pipeline of the at least one unit directly to the array (see page 11, last paragraph; see also page 25, Fig. 3.1 (a), wherein the processor controls the array; see also description of the figure in page 37).

27. As per claim 33, Rosenbluth further teaches a method according to claim 7, wherein a packet oriented data bus in the array is configurable (see Rosenbluth, Fig. 3). The motivation utilized in the combination of claim 7, super, applies equally as well to claim 33.

28. As per claim 34, Hauser in view of Rosenbluth further teaches a method according to claim 33, wherein the packet oriented data bus in the array is reconfigurable (see Hauser, wherein all the components of the array are reconfigurable including the buses, see reconfigurable array in page 25 and also see figure 3 of Rosenbluth).

29. As per claim 35, Hauser in view of Rosenbluth further teaches method according to claim 33, wherein the packet oriented data bus in the array is reconfigurable at runtime (see Hauser, page 25, Fig. 3.1(b) and code in page 111, wherein the array is dynamically reconfigurable; see also Fig. 3 or Rosenbluth).

30. As per claim 36, Hauser further teaches a method according to claim 7, wherein the cells are configurable (see figure in page 62 and figure in page 25 'reconfigurable array').

31. As per claim 37, Rosenbluth further teaches a method according to claim 7, wherein the packet oriented network is configurable (see fig. 3 of Rosenbluth).

32. As per claim 38, Hauser further teaches a method according to claim 7, wherein the cells are runtime reconfigurable (see Hauser, page 25, Fig. 3.1(b) and the code in page 111, wherein the array is dynamically reconfigurable).

33. As per claim 39, Hauser in view of Rosenbluth further teaches a method according to claim 7, wherein the packet oriented network is runtime reconfigurable (see Hauser, page 25, Fig. 3.1(b) and code in page 111, wherein the components is dynamically reconfigurable; see also Fig. 3 or Rosenbluth).

34. As per claim 40, Hauser further teaches the method according to claim 1, wherein the instructions are issued from the instruction pipeline of the at least one unit directly to the array (see page 11, last paragraph; see also page 25, Fig. 3.1 (a), wherein the processor controls the array; see also description of the figure in page 37).

35. As per claim 41, Hauser further teaches the method according to claim 1, wherein the at least one unit and the array exchange data via a joint D-cache (see page 30, paragraph beginning with 'With configuration sizes…" and last paragraph in page 37; see also figure in page 61, wherein both the array and the processor retrieves and write data to the cache).

36. As per claim 42, Hauser further teaches the method according to claim 1, wherein the at least one unit transmits data directly from within the data path to the array (see page 11, last paragraph; see also page 25, Fig. 3.1 (a), wherein the processor controls the array; see also description of the figure in page 37).

37. As per claim 43, Hauser further teaches a method according to claim 1, wherein the at least one unit transmits data directly from a Write stage of the data path to the array (see page 11, last paragraph; see also page 25, Fig. 3.1 (a), wherein the processor controls the array; see description of the figure in page 37; also see figure in page 61).

38. As per claim 44, Hauser further teaches a method according to claim 1, wherein the array transmits data directly into the data path of the at least one unit (see page 11, last paragraph; see also page 25, Fig. 3.1 (a), wherein the processor controls the array; see also description of the figure in page 37).

39. As per claim 45, Hauser further teaches a method according to claim 1, wherein the array transmits data directly into an Execute stage of the data path of the at least one unit (see page 11, last paragraph; see also page 25, Fig. 3.1 (a), wherein the processor controls the array; see description of the figure in page 37; also see figure in page 61).

40. As per claim 46, Hauser further teaches a method according to claim 1, wherein the array is accessible as a register file in the at least one unit (see Fig. 3.1(a), wherein the array is accessed as a component within the processor; see also Figure shown in page 61).

41. As per claim 47, Hauser further teaches a method according to claim 1, wherein the array is accessible in parallel to the register file in the at least one unit (see Fig. 3.1(a)).

42. As per claim 48, Hauser further teaches a method according to claim 1, wherein data is transferred between the at least on unit and the array via FIFOs (see Fig. 3.1(b) wherein a register or a FIFO is between the array and the processor, see also interface in page 37).

43. As per claim 49, Hauser further teaches a method according to claim 1, wherein data is transferred between the at least on unit and the array via dual-clock FIFOs (see page 34, section 3.1.5, as well known in the art, having single clock or dual clock would be depend on the application or design. Therefore, it would have been an obvious matter of design choice to have the data transferred or derived using dual clock).

44. As per claims 50-56, they are rejected for the same reasoning set forth above in claims 33-39.

45. As per claim 57, Hauser teaches a method of processing data comprising the step of:

coupling:

(a) at least one unit adapted for processing data in a sequential manner and comprising an instruction pipeline (see page 11, 'master processor'; see also page 25, Fig. 3.1(b), wherein the pipeline of the processor is sequential); and

(b) an array that:

adapted for processing data (page 25, Fig. 3.1(b) 'reconfigurable array'; see also page 11 'slave reconfigurable device');

comprises a plurality of data processing cells (see page 12, Fig. 2.6) having programmable coarse grained arithmetic logic units (ALUs) (see figures in page 62; see figure shown in page 64; see also page 52, section 3.2.3 for more details);

and

processes at least one program that is compiled from a high-level language (the instructions for processing are compiled from a high-level language such as C; see page 20, paragraph beginning with 'Probably the earliest…'; see also figure in page 108 and the 'software tools');

wherein the at least one unit is operable independently (see page 11, 'master processor'; see also page 25, Fig. 3.1(b)) of the array and transmits data directly from within a data path of the at least one unit to the array (see page 11, last paragraph; see

also page 25, Fig. 3.1, wherein the processor controls the array; see also description of the figure in page 37).

Hauser shows network communication between the main processor and the array but did not explicitly show that the array comprises a packet-oriented network communicating values in packets. However, the secondary reference (Rosenbluth) teaches an array (Rosenbluth Fig. 1, element 20's 'MEs') comprises a packet-oriented network communicating values in packets (see Rosenbluth, Fig. 3, packet processing; see also col. 2, lines 36-54), for the purpose of processing network packets between processors or a network of functional units with a maximum efficiency;

Accordingly, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teachings of Rosenbluth in the invention of Hauser, for the purpose of efficiently transmitting packets between functional units or processors within a network and significantly improving performances when transmitting in packets rather than in single instructions.

46. As per claims 58-59, they are rejected for the same reasons set forth above in claims 31-32.

47. As per claims 60-63, they are rejected for the same reasons set forth above in claims 43-46.

48. As per claims 64-65, they are rejected for the same reasons set forth above in claims 48-49.

49. As per claim 66, it is rejected for the same reasons set forth above in claim 47.

50. As per claims 67-72, they are rejected for the same reasons set forth above in claims 51-56.

51. Claims 5-6 and 21-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dawes U.S. Patent No. 4,967,340 (hereinafter Dawes) in view of Lin U.S. Patent No. 7,043,416 (hereinafter Lin).

52. As per claim 5, Dawes further teach a device for processing data comprising:

at least one unit adapted for processing data in a sequential manner (see Dawes, Fig. 2, element 48); and

an array adapted for processing data comprising a configurable network and a plurality of data processing cells that are configurable in their function (see Dawes, Fig. 2, elements 28, 50 and abstract);

wherein:

the array is coupled to the instruction pipeline (see Dawes, Fig. 2, wherein element 48 is coupled into element 28 “array processor”; For the pipeline, see Lin as explained below), the coupling of the array to the instruction pipeline including

controlling configurations by the instruction pipeline (see Dawes, Fig. 2, and abstract, wherein the CPU element 48 controls the configuration); and

the at least one unit is operable independently of the array (see Dawes, Fig. 2, element 48, which operates independent of the array processor element 28);

With regards to the instruction pipeline, although one of ordinary skill in the art at the time of the invention was made would understand that element 48 of Dawes comprises an instruction pipeline for processing instructions. It's common that modern processors have instruction pipelines in order to increase throughput. Applicant's indicated this in the specification as well, wherein RISC cores employ instruction pipelines to maximize throughput.

However, since Dawes was not explicit on the instruction pipelining, the examiner introduced a secondary reference (Lin) that explicitly shows a processor comprising an instruction pipeline (see Lin, Fig. 3 and col. 4, lines 62-67), for the purpose of maximizing throughput and increasing speed (see Lin, col. 4, lines 62-67 and col. 1, lines 2-41).

Thus, since it would have been obvious to one of ordinary skill in the relevant art to have a pipeline in the CPU (element 48) of Dawes, the array would be coupled to the instruction pipeline since it's coupled to the CPU (element 48) (see also Dawes, Fig. 2, wherein element 48 is coupled to element 28 "array processor").

Accordingly, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teachings of Lin in the invention of Dawes,

for the purpose of enhancing speed and increasing throughput thus significantly improving performances.

53. As per claim 6, Dawes further teaches the device according to claim 5, wherein at least one of:

at least one data path is provided between the array and the at least one unit, the at least one data path comprising at least one FIFO that allows at least one of (i) a coupling between the at least one unit and the array that is not synchronous and (ii) a data processing within the at least one unit and the array that is not synchronous (see Dawes, abstract and col. 5, lines 27-64; see also rejection of claim 2 for more details); and

data is transferred by at least one of extracting data directly from and inserting data directly into a data path of at least one of the at least one unit and the array (see Dawes, col. 5, lines 40-64).

54. As per claim 21, Dawes further teaches a device according to claim 5, wherein the at least one unit includes at least one of a CPU, a von-Neumann-Processor, and a microcontroller (see Fig. 2, element 48).

55. As per claim 22, Dawes further teaches a device according to claim 5, wherein the array includes a runtime and reconfigurable data processor (see abstract and col. 2, lines 2-24).

56. As per claim 23, Dawes further teaches a device according to claim 6, wherein the at least one data path includes at least one local memory connected to the unit as cache and connected to the array (see abstract and col. 5, lines 27-64).

57. As per claim 24, Dawes further teaches a method according to claim 23, wherein the at least one local memory includes an internal RAM (IRAM) (see col. 2, lines 12-24).

58. As per claim 25, Dawes and Lin further teach a device according to claim 5, wherein configuration information for the array is issued by the instruction pipeline (see Dawes, abstract and col. 5, lines 27-64; For the pipeline, see Lin as explained in claim 1).

59. As per claim 26, Dawes further teaches a device according to claim 25, wherein the configuration information is buffered in at least one FIFO so as to allow for at least one of (i) a coupling between the at least one unit and the array that is not synchronous and (ii) a data processing within the at least one unit and the array that is not synchronous (see Dawes, abstract and col. 5, lines 27-64; see rejection of claim 2 for more details).

60. As per claim 27, Dawes further teaches a device according to claim 5, wherein the at least one unit supports multi-threading, and the array is connected as a thread unit (see col. 5, lines 27-64).

61. As per claim 28, Dawes further teaches a device according to claim 27, wherein the array operates synchronously to the unit (see abstract and col. 5, lines 27-64).

Response to Arguments

62. Applicant's arguments with respect to claims 1-4, 7, 12-20 and 29-72 have been considered but are moot in view of the new grounds of rejection.

63. With respect to claim 5, applicant's arguments have been fully considered but they are not persuasive.

64. *Applicant argues that Dawes failed to show that the array is coupled by an instruction pipeline of at least one unit, wherein the instruction pipeline controls configuration*.

The examiner respectfully disagrees. With regards to the instruction pipeline, although one of ordinary skill in the art at the time of the invention was made would understand that element 48 of Dawes comprises an instruction pipeline for processing instructions. It's common that modern processors have instruction pipelines in order to

increase throughput. Applicant's indicated this in the specification as well, wherein RISC cores employ instruction pipelines to maximize throughput.

However, since Dawes was not explicit on the instruction pipelining, the examiner introduced a secondary reference (Lin) that explicitly shows a processor comprising an instruction pipeline (see Lin, Fig. 3 and col. 4, lines 62-67), for the purpose of maximizing throughput and increasing speed (see Lin, col. 4, lines 62-67 and col. 1, lines 2-41).

Thus, since it would have been obvious to one of ordinary skill in the relevant art to have a pipeline in the CPU (element 48) of Dawes, the array would be coupled to the instruction pipeline since it's coupled to the CPU (element 48) (see also Dawes, Fig. 2, wherein element 48 is coupled to element 28 "array processor").

Accordingly, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teachings of Lin in the invention of Dawes, for the purpose of enhancing speed and increasing throughput thus significantly improving performances.

With respect to controlling configuration, see Dawes, Fig. 2, and abstract, wherein the CPU element 48 controls the configurations of the array. Therefore the argued claims stand as previously rejected.

Conclusion

65. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

66. Any inquiry concerning this communication or earlier communications from the examiner should be directed to IDRISS N. ALROBAYE whose telephone number is (571)270-1023. The examiner can normally be reached on Mon-Fri from 8:00 to 4:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 571-272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eddie P Chan/
Supervisory Patent Examiner, Art Unit 2183

Idriss Alrobaye
AU 2183
(571) 270-1023